

MAVEN EXTERNAL DEPENDENCIES

http://www.tutorialspoint.com/maven/maven_external_dependencies.htm

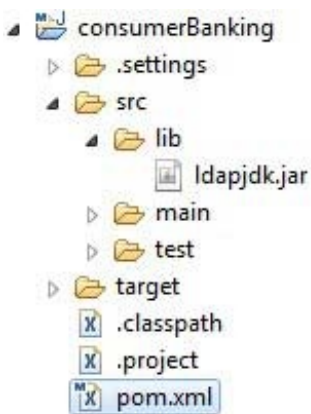
Copyright © tutorialspoint.com

Now as you know Maven does the dependency management using concept of [Maven Repositories](#). But what happens if dependency is not available in any of remote repositories and central repository? Maven provides answer for such scenario using concept of **External Dependency**.

For an example, let us do the following changes to project created in [Maven Creating Project](#) section.

- Add **lib** folder to src folder
- Copy any jar into the lib folder. We've used **ldapjdk.jar**, which is a helper library for LDAP operations.

Now our project structure should look like following:



Here you are having your own library specific to project, which is very usual case and it can contain jars which may not be available in any repository for maven to download from. If your code is using this library with Maven then Maven build will fail because it cannot download or refer to this library during compilation phase.

To handle the situation, let's add this external dependency to maven **pom.xml** using following way.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.companyname.bank</groupId>
  <artifactId>consumerBanking</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>consumerBanking</name>
  <url>http://maven.apache.org</url>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>ldapjdk</groupId>
      <artifactId>ldapjdk</artifactId>
      <scope>system</scope>
      <version>1.0</version>
      <systemPath>${basedir}\src\lib\ldapjdk.jar</systemPath>
    </dependency>
  </dependencies>
</project>
```

```
        </dependency>
    </dependencies>

</project>
```

Look at the second dependency element under dependencies in above example which clears following key concepts about **External Dependency**.

- External dependencies (library jar location) can be configured in pom.xml in same way as other dependencies.
- Specify groupId same as name of the library.
- Specify artifactId same as name of the library.
- Specify scope as system.
- Specify system path relative to project location.

Hope now you are clear about external dependencies and you will be able to specify external dependencies in your Maven project.